Effective Path Identification Protocol for Wireless Mesh Networks

Avinash Bashyam[1], Anam Chenchu Jeevan[2], Sai Kiran Tripuramalla[3] and Suresh Babu Sanka[4]
[1,2,3,4] Information Science & Technology, KL University, Vaddeswaram, India.
Email: y6it216@klce.ac.in[1]
Email: {y6it224[2] ,y6it282[3],y6it314[4]}@klce.ac.in

***Abstract*—Wireless Mesh Networks (WMNs) have emerged as a key technology for next-generation wireless networking. Routing is a key factor for transfer of packets from source to destination. SrcRR is widely used protocol for transferring packets from source to destination. This protocol often uses Dijkstra's algorithm on its link state database to find the next alternative path to the destination when ever the ETX metric of the link changes.This is a time consuming process if the ETX metric of the links are changing frequently. So this paper eliminates the use of Dijkistra's algorithm and uses the a search operation for finding the best paths.**

***Index Terms* --Mesh networks, Routing, Reactive, Dijkstra.**

I. INTRODUCTION

In Wireless mesh networks (WMNs) the communication is through radio nodes organized in the mesh topology. The primary advantages of a WMN lie in its inherent fault tolerance against network failures, simplicity of setting up a network, and the broadband capability. Although by definition a WMN is any wireless network having a network topology of either a partial or full mesh topology, practical WMNs are characterized by static wireless relay nodes providing a distributed infrastructure for mobile client nodes over a partial mesh topology. Due to the presence of partial mesh topology, a WMN utilize multihop relaying similar to an ad hoc wireless network.

The routing protocols in WMN are classified into reactive, proactive and hybrid protocols [2]. In case of reactive protocols the routes are established when ever required. Proactive protocols have routes already defined in their routing tables. Hybrid protocol is a combination of both reactive and proactive protocols. SrcRR protocol comes under reactive protocol and it an extension of DSR protocol [3].

The special routing metrics of WMN protocols are Expected number of Transmissions (ETX), Expected Transmission time (ETT), Weighted Cumulative ETT (WCETT)[1]. During the early stages, WMN used many of the Adhoc protocols for routing. But these protocols does not follow ETX,ETT,WCETT metrics, so it failed to achieve reliability, scalability, throughput, load balancing, congestion control over WMN. Section2 deals with description of SrcRR protocol and Section3 deals with the proposed idea and Section4 deals with Conclusion.

II. SrcRR ROUTING PROTOCOL

SrcRR is an extension of DSR. SrcRR mainly deals with throughput by considering link loss and transmission bit-rate and transient bursts. This protocol mainly deals with the ETX metric [6].

ETX is the expected transmissions required to transmit the data packet from one node to another. ETX continuously measures the loss rate in both directions between each node and its neighbors using periodic broadcasts. It assigns each link a metric that estimates the number of times a packet will have to be transmitted before it (and the corresponding 802.11 ACK) are successfully received; thus the best link metric is one. The ETX route metric is the sum of the link metrics; thus ETX penalizes both long routes and routes that include links with high forward or reverse loss rates.

Every node running SrcRR maintains a link cache, which tracks the ETX metric values for links it has heard about recently. Whenever a change is made to the link cache, the node locally runs Dijkstra's weighted shortest-path algorithm on this database to find the current, minimum-metric routes to all other nodes in the network. To ensure only fresh information is used for routing, if a link metric has not been updated within 30 seconds it is dropped from the link cache [6].

When a node wants to send data to a node to which it does not have a route, it floods a route request. When a node receives a route request, it appends its own node ID, as well as the current ETX metric from the node from which it received the request, and rebroadcasts it. A node will always forward a given route request the first time it receives it. If it receives the same route request again over a different route, it will forward it again if the accumulated route metric is better than the best metric it has forwarded so far. This ensures that the target of the route request will receive the best routes.

When a node receives a route request for which it is the target, it reverses the accumulated route and uses this as the source-route for a route reply. When the original source node receives this reply, it adds each of the links to its link cache, and then source-routes data over the minimum-metric path to the destination. When a SrcRR node forwards a source-routed data packet, it updates its entry in the source route to contain the latest ETX metric for the link on which it received the packet.

DOI: 01.IJNS.01.03.145

This allows the source and destination to maintain up-to date link caches, and discover when a route's quality has declined enough that an alternate route would be better. In addition, each data packet includes a field to hold one extra link metric; a forwarding node will randomly, with probability 1/n, where n is the number of nodes in the route, that field with the ETX metric to one of its neighbors. This allows the source and destination to learn of the existence and metric of some alternate links. As with all changes to the link cache, this prompts re-computation of all the best routes using Dijkstra's algorithm. All query and data packets contain ETX metrics for the links they have traversed so far. Any node that receives such a packet (including forwarding nodes) copies those metrics to its link cache.

Baseline SrcRR broadcasts a 300-byte ETX probe packet at randomized intervals averaging every ten seconds. ETX measures the loss rate from each neighbor by counting the fraction of probes received over the last three minutes (18 probes).

SrcRR is independent of IP, and operates at a lower layer. SrcRR uses 32-bit addresses; in the usual case in which it is carrying IP packets, SrcRR use IP addresses in its headers. A SrcRR node maintains a mapping from SrcRR 32-bit addresses to 48-bit 802.11 MAC addresses, derived implicitly from SrcRR query broadcasts.

A. *Advantages:*

- Finds routes with high throughput rates.
- The use of ETX metric penalizes ETX both long routes and routes that include links with high forward or reverse loss.

B. *Disadvantages :*

- SrcRR is not likely to scale to more than a few hundred nodes. As SrcRR uses dijkistra algorithm every change in the network topology allows the nodes to run the algorithm [7].
- A node forwards a query if it has not seen the query before, or if the query's total route metric is better (lower) than the best instance of the query the node has yet seen. This increases the amount of query traffic.

III. PROPOSED IDEA

SrcRR mainly uses weighted Diskistra algorithm among all the paths to find the path with less ETX metric as the path to destination.

But this algorithm lacks scalability and takes some time to run Dijkstra algorithm when the number of nodes in the network is more than 500. This is because as the number of nodes of in the network increases, the number of paths to reach the destination also increases.

When there is a change in the ETX metric on the current path to the destination, then the source will run the local Dijkstra algorithm on the link cache to find the next best path. So when there are more number of nodes and the link metric changes frequently then running the Dijkstra algorithm to find the next path will consume a lot of time.

The proposed idea improves the algorithm by using the search operation on the linked list to find the best path instead of using the Dijkstra algorithm to find the best path.

Step 1 **:** When ever there is a change in the ETX metric along the link , then the node node must include this information in the forwarding packet, ie :the link where the ETX metric is changed.

Step 2**:** Then that node has to include the ETX metric of the adjacent nodes whose ETX metric is less than the previous ETX metric.

Step 3**:** When ever a new nodes adds into the network it must calculate the ETX metric with the adjacent node and this information must be included in the forwarding packet by the intermediate node.

Step 4**:** Consider all links upto where the link metric has changed. Instead of using Dijkstra algorithm use this path and do search operation in the link state database to find the similar paths.

Step 5**:** As we get the nodes from the ACK packet. Do the search operation based on these nodes and find the best path to destination.

Consider an example network:

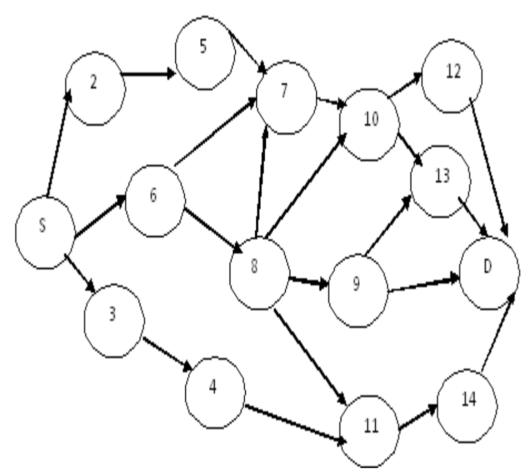

Figure 1: Example Network

The best paths to the destination that has minimum ETX metric values are:

1. S-6-8-9-10-D
2. S-6-7-10-13-D
3. S-6-8-11-14-D
4. S-3-4-11-14-D
5. S-6-8-10-13-D

DOI: 01.IJNS.01.03.145

Among these paths consider the best path as S-6-8-9-10-D. Consider this as the path with less ETX metric. If the ETX metric of the link 8-9 has been increased and this information is sent to the source along with the other alternative links from the node 8.

Previous case the algorithm should be applied for the link state database and should find the best path .but in this case the search operation is sufficient.

The alternative links to the destination from node 8 are 7, 10, 11. This information is sent to the source. The source instead of running the Dijkstra algorithm does the search operation by considering the path S-6-8.

The next alternative path to the destination from node 8 are S-6-8-11-14-D and S-6-8-10-13-D . Among these 2 paths consider the best path .

So this type of finding the path will eliminate the use of Dijkstra algorithm frequently by the source. This alternative path is the effective path to the network. But the throughput may change a little bit.

IV. CONCLUSION

Effective path computation is most challenging factor that has to be considered in all the protocols that have been used till now.. The above proposed idea eliminates the drawback of SrcRR protocol, effectively finds the path when the ETX metric of the node has been changed.

When ever the nodes are mobile use of search operation will reduce the lot of computation work to find the next path in very easy manner. So use of search operation makes the algorithm to work efficiently.

In the above proposed protocol throughput effects in a slight manner. So we try to improve that throughput and we also try to improve scalability by using the effective routing metrics for the algorithm computation.

REFERENCES

[1] Wireless mesh networks architectures and protocols by Yan Zhang.

[2] Wireless Mesh Routing Protocols For Health Communication Systems,11th Panhellenic Conference in Informatics.

[3] Wireless mesh networks : a survey, by Ian F Akyildiz Xudong Wang Computer Networks 47 (2007) pg 445-487.

[4] Performance Study of Routing Protocols for Wireless Mesh Networks, by Anna Zakrzewska, Leszek Koszałka, Iwona Poźniak-Koszałka19th International Conference on Systems Engineering.

[5] Routing Metrics and Protocols for Wireless Mesh Networks,IEEE networks.

[6] SrcRR: A High Throughput Routing Protocol for 802.11 Mesh Networks (DRAFT) by Daniel Aguayo, John Bicket, Robert Morris.

[7] MIT Roofnet Implementation overview.

[8] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In Proceedings of the 9th ACM International Conference on Mobile Computing and Networking (MobiCom '03), San Diego, California, September 2003.

DOI: 01.IJNS.01.03.145